Efficient Algorithms for Characteristic Wireless Power Transfer Problems in Sensor Networks

Sotiris Nikoletseas[1,2] and Theofanis P. Raptis[1,2(✉)]

[1] Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
nikole@cti.gr
[2] Computer Technology Institute and Press "Diophantus", Patras, Greece
traptis@ceid.upatras.gr

Abstract. In Wireless Rechargeable Sensor Networks, one or more special mobile entities (called the Mobile Chargers) traverse the network and wirelessly replenish the energy of sensor nodes, using wireless power transfer technology. In this paper, we present some state of the art algorithms that apply to characteristic problems in such networks, namely efficient use of wireless power transfer using i) one Mobile Charger, ii) multiple Mobile Chargers, iii) collaborative mobile charging.

Keywords: Wireless Power Transfer · Wireless Sensor Networks · Energy Management · Distributed Algorithms · Mobility

1 Introduction and Model

Wireless Rechargeable Sensor Networks have recently attracted much research interest (e.g. [2], [3], [4], [6]). In this paper, we review some state of the art algorithms that apply to characteristic problems in Wireless Rechargeable Sensor Networks: (a) how can we use a single Mobile Charger efficiently ([1]), (b) how can we use multiple Mobile Chargers efficiently ([5]) and (c) how can we use multiple Mobile Chargers, capable of charging each other, efficiently.

Our model features three types of devices: stationary sensors, Mobile Chargers and one stationary Sink. We assume that there are N sensors of wireless communication range r distributed at random in a circular area of radius R. We virtually divide the network into Slices, the number of which is equal to the number of the Mobile Chargers. K Mobile Chargers initially deployed at coordinates $(x, y) = (\frac{R}{2}\cos(\frac{\pi}{K}(2j-1)), \frac{R}{2}\sin(\frac{\pi}{K}(2j-1)))$ of the circular area, where

S. Nikoletseas: Research supported by the European Social Fund (ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thalis-DISFER, Investing in knowledge society through the European Social Fund.

T.P. Raptis: Research supported by the EU/FIRE IoT Lab project - STREP ICT-610477.

N. Mitton et al. (Eds.): ADHOCNETS 2014, LNICST 140, pp. 204–215, 2014.
DOI: 10.1007/978-3-319-13329-4_18

$j = 1, 2, ..., K-1$ (one Mobile Charger per Slice). In the case where $K = 1$, the Mobile Charger is initially deployed at the centre of the circular area. The Sink lies at the centre of the circular area. In our model we assume that the Mobile Chargers do not perform any data gathering process.

We denote by E_{total} the total available energy in the network. Initially, $E_{total} = E_{sensors} + E_{MC}(t_{init})$ where $E_{sensors}$ is the amount of energy shared among the sensor nodes and $E_{MC}(t_{init})$ is the total amount of energy that the Mobile Chargers have and may deliver to the network by charging sensor nodes. The maximum amount of energy that a single node and a single charger may store is $E^{\max}_{sensor}$ and $E^{\max}_{MC}$ respectively. Energy is split among the sensor nodes and the chargers as follows: $E^{\max}_{sensor} = \frac{E_{sensors}}{N}$ and $E^{\max}_{MC} = \frac{E_{MC}(t_{init})}{K}$. We denote as E_i and E_j the residual energy of sensor node i and Mobile Charger j respectively.

In our model the charging is performed point-to-point, i.e. only one sensor may be charged at a time from a Mobile Charger by approaching it at a very close distance so that the charging process has maximum efficiency. The time that elapses while the Mobile Charger moves from one sensor to another is considered to be very small when compared to the charging time; still the trajectory followed (and particularly its length) is of interest to us, since it may capture diverse cost aspects, like gas or electric power needed for charger movement. We assume that the charging time is equal for every sensor and independent of its battery status.

We assume a quite heterogeneous data generation model. Each sensor node chooses independently a relative data generation rate $\lambda_i \in [a, b]$ (where a, b constant values) according to the uniform distribution $\mathcal{U}[a, b]$. Values of λ_i close to a imply low data generation rate and values close to b imply high data generation rate. The routing protocol operates at the network layer, so we are assuming appropriate underlying data-link, MAC and physical layers.

1.1 Energy/Flow Criticality

In order to develop efficient algorithms for the Mobile Charger and address the corresponding trade-offs, we introduce *an attribute that captures a node's "importance"* in the network, under any given routing protocol. This attribute relies on two factors, (a) the *traffic served* by the node and (b) the *energy consumed* by the node.

The need for combining these two factors emerges from the fact that the traffic served by a node captures different aspects than its energy consumption rate. A node may consume a large amount of energy either because it serves a high network flow, or because its transmissions have high cost (e.g. long ranged transmissions) (or both). The purpose of the attribute is to indirectly prioritize the nodes according to their flow rate and energy consumption; a node serving high traffic and/or having low residual energy should be charged at higher energy level.

We denote as $c_i(t)$ the *energy/flow criticality* (also referred as criticality for simplicity) of node v_i at time t, with $c_i(t) = f_i(t) \cdot \rho_i(t)$. Given the time t_{MC}

when the last charging of the node occurred,

$$f_i(t) = 1 - \frac{\textit{generation rate of node } v_i}{\textit{traffic rate of } v_i \textit{ since } t_{MC}} = 1 - \frac{\lambda_i}{\lambda_i + \frac{m_i(t)}{t - t_{MC}}}$$

is the *normalized traffic flow* served by node v_i, where $m_i(t)$ *is amount of traffic (number of messages) that* v_i *has processed (received and forwarded) towards the Sink by time* t *since time* t_{MC}, and

$$\rho_i(t) = \frac{\textit{energy consumed since last charging}}{\textit{max node energy since } t_{MC}} = \frac{E_i(t_{MC}) - E_i(t)}{E_i(t_{MC})} = 1 - \frac{E_i(t)}{E_i(t_{MC})}$$

is the *normalized energy consumption* by time t, since the last charging. The criticality is thus a number in $[0, 1]$ which captures the importance of a given node by taking into account its flow rate, its energy consumption, its possible special role in the network and its influence to the routing protocol; nodes serving high traffic (large $m_i(t)$) and/or having consumed a lot of energy (low $E_i(t)$) have high criticality $c_i(t)$ at time t and *are "prioritized"* by the Mobile Charger.

2 Algorithms Using a Single Mobile Charger

2.1 Global Knowledge Algorithm GK

The global-knowledge charger we suggest is an on-line method that uses criticality as a ranking function. In each round, the charger moves to the sensor that minimizes the product of the negation of each node's criticality times its distance from the current position of the Mobile Charger. More specifically, in each moving step the GK minimizes the product

$$\min_i \left\{ (2 - c_i(t)) \cdot \left(1 + \frac{dist_i}{2R} \right) \right\}$$

where $dist_i$ is the distance of each sensor from the Mobile Charger and D is the network radius, with the minimum taken over all sensors in the network (or at least a large part of it). In other words, *this algorithm prioritizes nodes with high criticality and small distance to the Mobile Charger*. Since this algorithm requires *a global knowledge of the state of the network*, it is expected to outperform all other strategies that use only local or limited network information, thus somehow representing an on-line centralized performance upper bound. However, it would not be suitable for large scale networks as it introduces great communication overhead (i.e. every node has to propagate its criticality to the Mobile Charger) and does not scale well with network size.

2.2 Limited Reporting Algorithm LR

The Sink is informed about the status of some representative nodes scattered throughout the network and is able to provide the Mobile Charger with some

guidance. In other words, this algorithm distributively and efficiently "simulates" the global knowledge algorithm. We assume that the Sink can transmit to the Mobile Charger wherever in the network the latter might be. The algorithm follows a limited reporting strategy, since it exploits information from the whole network area but from a limited number of nodes. The nodes of each Slice periodically run a small computation overhead algorithm in order to elect some special nodes, the *reporters* of the Slice; in particular, each node becomes a reporter independently with some appropriate probability (thus, the number of reporters is binomially distributed). The reporters act as the representatives of their Slice and their task is the briefing of the Sink about their criticality.

The *percentage of the nodes that will act as reporters* brings off a trade-off between the representation granularity of the network and the communication overhead on each message propagated in the network. If we set a large percentage of reporters, the Sink will have a more detailed knowledge of each Slice's overall criticality but the message overhead will highly increase, since each message should carry the Slice reporter's current criticality. On the contrary, if we set a small percentage of reporters, the overhead will be tolerable, but the representation of a Slice will be less detailed.

In order to maintain *a small set of reporters* for each Slice (for communication overhead purposes) we propose that Slice i which contains n_i nodes elects $\kappa_i = \frac{n_i}{N} \cdot \kappa_{total}$ reporters, with the global number of reporters being

$$\kappa_{total} = h \frac{R}{r} \log N, \text{ where } h = 1 - \frac{a}{b}$$

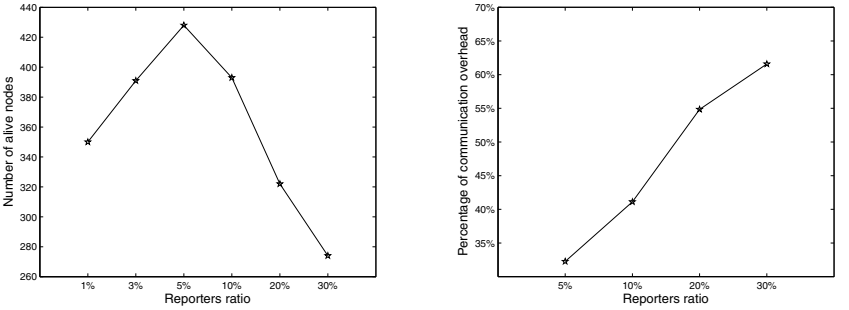
is a network density heterogeneity parameter. Clearly, a highly heterogeneous deployment (large b compared to a) will necessitate a higher number κ_{total} of reporters. Also, κ_{total} must be large in large networks with many sensors. Each node periodically with probability p_i becomes a reporter. In order to have an expected number of κ_i reporters in Slice i we need:

$$\kappa_i = n_i \cdot p_i \Rightarrow p_i = \frac{\frac{n_i}{N} \cdot \kappa_{total}}{n_i} \Rightarrow p_i = \frac{\kappa_{total}}{N}$$

The reporter selection is meant to happen in a local and distributed manner, i.e. each node becomes a reporter with the above suitably, independently chosen probability. This random independent generation of reporters is captured by Bernoulli trials (one per node) i.e. a binomial distribution. In order to figure out possible good values for κ_{total} that maximize the LR performance we carry out a comparison operating the protocol between several reporter numbers (Fig. 1).

2.3 Reactive Trajectory Algorithm RT

In this algorithm, a node v_i is propagating an *alert message* to its neighbours each time its energy drops below a set of some crucial limits. The messages are propagated for some hops and are stored at every node passed, in order for a *tree structure* rooted at v_i to be formed that can be *detected by the Mobile Charger*



(a) Alive nodes. (b) Overhead.

Fig. 1. Alive nodes and communication overhead for various κ_{total} values in LR, after 6000 generated events

when passing through some tree node. Every node can root a tree and the strategy followed (towards a small tree management overhead) is the maintenance of a small tree degree with a larger tree depth.

The tree that is formed for each node is gradually growing, in an analogous way to the criticality of the root node, as the gradual increase of a node's criticality is an indication of either high traffic or high energy consumption. *We use criticality as a measure of the gradual expansion of the tree*, since its value depicts both the importance of the node in the network and its energy consumption rate. We propose a strategy of message propagations that aims at covering a relatively large area of the network, while keeping energy consumption due to communication overhead low.

More specifically, each node v_i can alter among $\left\lceil \log\left(N\frac{R}{r}\right)\right\rceil$ alert levels which determine the characteristics of the v_i's rooted tree. We denote as al_i the current *alert level* of node v_i. The tree rooted at v_i is formed in a way that the $degree = al_i - 1$ and the $depth = 2^{al_i-1} - 1$. The duration of each successive alert level is increased by a constant ratio from the previous level:

$$al_i = \begin{cases} 1 & \text{if } 0 \le c_i(t) < 0.5 \\ 2 & \text{if } 0.5 \le c_i(t) < 0.75 \\ \vdots & \vdots \\ \left\lceil \log\left(N\frac{R}{r}\right)\right\rceil & \text{if } 1 - \frac{1}{2^{\left\lceil \log\left(N\frac{R}{r}\right)\right\rceil - 1}} \le c_i(t) < 1 \end{cases}$$

$$= \left\{ \mu \mid \mu \in \left[1, 2, ..., \left\lceil \log\left(N\frac{R}{r}\right)\right\rceil\right]\right\}$$

$$\text{with } 1 - \frac{1}{2^{\mu-1}} \le c_i(t) < 1 - \frac{1}{2^{\mu}} \text{ where } 1 - \frac{1}{2^{\mu-1}} = \sum_{j=1}^{\mu-1} \frac{1}{2^j}, 1 - \frac{1}{2^{\mu}} = \sum_{j=1}^{\mu} \frac{1}{2^j}.$$

The tree management procedure aims at providing a high level information about the local trees' state and at the same time at maintaining the node memory reservation at relatively low levels. For this reason, nodes store information solely about their parent nodes in emerging tree structures (i.e. one record per parent). Node v_i which is already a tree member, may receive alert messages coming from nodes that belong to other trees. In this case, v_i stores the received alert messages from surrounding parents, the number of which is at most equal to the number of v_i's neighbors (since a parent node of v_i can only be in its transmission range). Nodes that participate in multiple trees, propagate messages concerning solely the highest alert level and redirect the Mobile charger (when the latter is near the current node) to follow the highest alert level tree links. In short, each node can participate in multiple trees, reserves memory at most equal to the number of its neighbors, propagates messages about the highest priority tree and redirects the Mobile Charger to it.

The Mobile Charger alters its state between a *patrol mode* and a *charging mode*. When in patrol mode, it follows a spiral patrol trajectory centred at the Sink and does not charge any nodes until notified that the area traversed is low on energy. When so notified by a node in such an area, it pauses the patrol mode and enters the charging mode, in which it follows a different trajectory in order to accomplish the charging process in this area. If the Mobile Charger detects simultaneously different trees, then by a check on the *depth* of each structure it can decide which is the most critical. After the completion of the charging process the Mobile Charger resumes the patrol mode.

3 Algorithms Using Multiple Mobile Chargers

3.1 Centralized Coordination Algorithm CC

The CC protocol performs centralized coordination among the chargers and assumes no knowledge on the network. In particular, the coordination process is able to use information from all Mobile Chargers (energy status, position etc), but is agnostic of the underlying network and sensor nodes attributes (energy status, position etc.). This approach virtually partitions the network elements in two completely separate levels, the Mobile Chargers level and the sensor nodes level.

Coordination phase. Each Mobile Charger is assigned to a network region. Since the initial charger deployment coordinates are $(x, y) = (\frac{R}{2}\cos(\frac{\pi}{K}(2j - 1)), \frac{R}{2}\sin(\frac{\pi}{K}(2j - 1)))$, where $j = 1, 2, ..., K$, we can split the network area in Slices, with one charger assigned to each Slice. When the coordination process is initialized, the region of each charger is computed. Each charger should be assigned to a region of size analogous to its current energy level, so that the energy dissipation among the chargers is balanced. In order to compute the size of the region of charger j, it suffices to compute the central angle ϕ_j corresponding to the charger's Slice. In particular

$$\phi_j = 2\pi \cdot \frac{E_j}{\sum_{j=1}^{K} E_j}, \text{ where } \sum_{j=1}^{K} \phi_j = 2\pi.$$

Charging phase. During this phase, charger j traverses the network region it is assigned to (Slice defined by angle ϕ_j) and charges the corresponding sensor nodes. The CC algorithm assumes no knowledge on the network. For this reason the path followed by the Mobile Charger is restricted to several naive alternatives. In our approach we use a "blind" scanning of the region where the Mobile Charger starts form the Sink and traverses an exhaustive path until it reaches the boundaries of the network area. The advantage of this movement is that due to its space filling attributes, the Mobile Charger covers the whole Slice and almost every node is charged, until the energy of the Mobile Charger is totally depleted. On the other hand, due to lack of knowledge, this movement is not adaptive, i.e. it does not take into account differences of the energy depletion rates of the network area caused by the underlying message propagation.

3.2 Distributed Coordination Algorithm DC

Coordination phase. The DC algorithm performs distributed coordination among chargers and assumes no network knowledge. We split the network area in Slices and assign one Slice per charger. Angle ϕ_j corresponds to the central angle of jth charger's Slice. The chargers distributively define their Slice limits (i.e. the two radii that define the Slice), according to the size of the region each one can handle, w.r.t. their energy status. Each charger can shift their right and left Slice limits resulting in either a widening or a shrinkage of the region of interest. This task is performed distributively and each region limit movement is determined through a cooperation of the two adjacent Mobile Chargers. A limit movement of j's region is expressed as a change of ϕ_j. The coordination process uses two critical charger parameters for definition the region of interest, the charger's current energy level E_j and the charger's energy consumption rate since the last coordination ρ_j. The change $\Delta\phi_j^l$ of ϕ_j for the left Slice limit and the change $\Delta\phi_j^r$ of ϕ_j for the right are defined by the following computations:

if $\min\{E_j, E_{j-1}\} = E_j$ **then**
  $\Delta\phi_j^l = -\phi_j \cdot \frac{|\rho_{j-1}-\rho_j|}{\max\{\rho_{j-1},\rho_j\}}$
else
  $\Delta\phi_j^l = \phi_{j-1} \cdot \frac{|\rho_{j-1}-\rho_j|}{\max\{\rho_{j-1},\rho_j\}}$
end if

if $\min\{E_j, E_{j+1}\} = E_j$ **then**
  $\Delta\phi_j^r = -\phi_j \cdot \frac{|\rho_j-\rho_{j+1}|}{\max\{\rho_j,\rho_{j+1}\}}$
else
  $\Delta\phi_j^r = \phi_{j+1} \cdot \frac{|\rho_j-\rho_{j+1}|}{\max\{\rho_j,\rho_{j+1}\}}$
end if

The new angle (denoted by ϕ'_j) is computed as $\phi'_j = \phi_j + \Delta\phi_j^l + \Delta\phi_j^r$. Note that, between two adjacent chargers j_1 and j_2, the change of their common slice limit is $\Delta\phi_{j_1}^r = -\Delta\phi_{j_2}^l$ so that the charger with the lower energy level provides its neighbor with a portion of its region of interest. Also, it is their energy level that determines which charger should reduce its region of interest and the energy consumption rate that determines the size of the reduced area. The size of the angle change is not computed by considering the energy levels of the two

chargers because energy consumption rate shows how quickly will this energy level be reduced. For example, if ρ_j is high then j's Slice is critical, causing a rapid reduction of E_j, independently of its current level.

Charging phase. Since this algorithm operates under the no knowledge assumption, the charging phase follows the same pattern with the CC algorithm (Slice scanning).

3.3 Centralized Coordination Global Knowledge Algorithm CCGK

The CCGK algorithm, similarly to the CC algorithm, performs centralized coordination. However, the assumption of global knowledge on the network further extends the Mobile Chargers' abilities. For this reason, it is expected to outperform all other strategies that use only local information, thus somehow representing a performance bound. The global knowledge assumption would be unrealistic for real large-scale networks, as it introduces large communication overhead (i.e. nodes and chargers have to propagate their status over large distances).

Coordination phase. Instead of using the same coordination process with the CC algorithm, we integrate the global knowledge assumption in the coordination phase. As a result, the network is not partitioned in two separate levels (Mobile Chargers, sensor nodes) and the Mobile Chargers are allowed to use network information during this phase. Each Mobile Charger is assigned to a network region. The region of interest of charger j is a cluster of nodes. Node i belongs to the cluster of charger

$$j' = \arg\min_j \left\{ \left(1 + \frac{dist_{ij}}{2R}\right) \cdot \left(2 - \frac{E_j}{E_{MC}^{\max}}\right) \right\}$$

where $dist_{ij}$ is the distance between node i and charger j. In other words, a node selects a charger which is close and with high amount of energy. Note that the centralized computation of the charger region in the CCGK algorithm is more powerful compared to other methods, since it uses information about the distance among every charger with every node.

Charging phase. The global knowledge charging phase we suggest uses energy and distance in a ranking function. In each round the charger moves to the sensor in the corresponding cluster, that minimizes the product of each node's energy times its distance from the current position of the Mobile Charger. More specifically, in each moving step the charger j charges node

$$i' = \arg\min_{i \in C_j} \left\{ \left(1 + \frac{dist_{ij}}{2R}\right) \cdot \left(1 + \frac{E_i}{E_{sensor}^{\max}}\right) \right\}.$$

In other words, this algorithm prioritizes nodes with low energy and small distance to the Mobile Charger.

3.4 Distributed Coordination Local Knowledge Algorithm DCLK

Coordination phase. The coordination phase follows the same pattern with the coordination phase of DC algorithm (distributed ϕ_j angle computation).
Charging phase. The DCLK algorithm operates with local knowledge assumption. The Slice corresponding to charger j is divided into k Sectors S_{jk} of the same width. Charger j prioritizes its Sectors w.r.t. high number of sensor nodes with low level of residual energy.

Definition 1. *$E_{jk}^{\min}$ is the lowest nodal residual energy level in the Sector S_{jk}.*

Definition 2. *$E_{jk}^{\min+\Delta}$ is an energy level close to $E_{jk}^{\min}$:*

$$E_{jk}^{\min+\Delta} = E_{jk}^{\min} + \delta \cdot \frac{E_{sensor}^{\max}}{E_{jk}^{\min}}, \delta \in (0,1).$$

Definition 3. *$N(S_{jk})$ is the number of nodes in Sector S_{jk} with residual energy between $E_{jk}^{\min}$ and $E_{\min+\Delta}^{jk}$:*

$$N(S_{jk}) = \sum_{e=E_{jk}^{\min}}^{E_{jk}^{\min+\Delta}} N(e)$$

where $N(e)$ is the number of nodes with energy level e.

Charger j charges Sector S_{jk} which maximizes the product $\max_{S_{jk}}\{N(S_{jk}) \cdot (E_{sensor}^{\max} - E_{jk}^{\min})\}$. The intuition behind this charging process is the grouping of nodes in each Slice and the selection of a critical group. A critical group is a Sector containing a large number of sensor nodes that require more energy than other nodes throughout the network.

4 Algorithms Using Hierarchical Collaborative Mobile Charging

4.1 Model Variation

In this case, Mobile Chargers can either charge nodes or charge other Mobile Chargers. Initially, $E_{total} = E_{sensors} + E_{MC}(t_{init}) + E_{SC}(t_{init})$, where $E_{sensors}$ is the total amount of energy shared among the sensor nodes, $E_{MC}(t_{init})$ is the total amount of energy shared among the Mobile Chargers and $E_{SC}(t_{init})$ is the total amount of energy shared among the Special Chargers. The maximum amount of energy that a single node, a single Mobile Charger and a single Special Charger may store is $E_{sensor}^{\max}$, $E_{MC}^{\max}$ and $E_{SC}^{\max}$ respectively. Energy is uniformly split among the sensor nodes and the chargers as follows: $E_{sensor}^{\max} = \frac{E_{sensors}}{N}$, $E_{MC}^{\max} = \frac{E_{MC}(t_{init})}{M}$ and $E_{SC}^{\max} = \frac{E_{SC}(t_{init})}{S}$.

At first, we deploy the sensor nodes uniformly in the circular network. Then, we divide our network into M equal sized Slices, one for each Mobile Charger. Thus, every Mobile Charger is responsible for charging nodes that belong to its Slice. We denote by D_j the set of sensor nodes that belong to Slice j, i.e. to the jth Mobile Charger's group. Finally, we divide the Mobile Chargers into S groups, one for each Special Charger. Thus, each Special Charger is responsible for charging the Mobile Chargers that belong to its group, denoted as C_k (for SC_k). Initially, these S groups are equally sized, i.e. $|\mathcal{C}_k| = \frac{M}{S}$ $(1 \leq k \leq S)$ and the Mobile Chargers that belong to each group are given by the following formula: $\mathcal{C}_k = \left\{j : j \in \left[(k-1)\frac{M}{S} + 1\ ,\ k\frac{M}{S}\right]\right\},\ (1 \leq k \leq S)$

These groups may change during the algorithm's coordination phase. More specifically, the Special Chargers communicate with each other and decide, according to their energy status, if they are still able to be in charge of the Mobile Chargers that belong to their group or they should delegate some of them to other Special Chargers.

4.2 1-Level Knowledge Distributed Coordination 1KDC

The 1KDC algorithm performs a distributed coordination among Special Chargers, i.e. every Special Charger can communicate with adjacent neighbors. Also, it assumes 1-level network knowledge, i.e. it can use information only about Mobile Chargers' energy status (and not about the sensors' which lie one level lower).

Coordination phase: In distributed coordination, we assume that a Special Charger knows which are the adjacent Mobile Chargers on the boundaries of its region. We call next the first Mobile Charger that belongs to the SC_{k+1} and previous the last Mobile Charger that belongs to SC_{k-1}. More specifically, $n_k = \min_{j \in C_{k+1}} \{j\}$: next Mobile Charger (belongs to SC_{k+1}) $p_k = \max_{j \in C_{k-1}} \{j\}$: previous Mobile Charger (belongs to SC_{k-1})

The Special Charger SC_k, in order to coordinate with each of its neighbors (SC_{k-1} and SC_{k+1}), calculates which of them has the highest energy supplies so as to charge the Mobile Chargers in its group and the additional Mobile Charger of its left or right neighbor. Thus, every Special Charger k estimates the residual energy in both cases (including a Mobile Charger of its left and right neighbor) by the following equations:

$$e_k^p = E_{SC_k} - \sum_{j \in \mathcal{C}_k} E_{MC_j}^{lack} - E_{MC_{p_k}}^{lack},\ \ e_k^n = E_{SC_k} - \sum_{j \in \mathcal{C}_k} E_{MC_j}^{lack} - E_{MC_{n_k}}^{lack}$$

where $E_{MC_j}^{lack} = E_{MC}^{\max} - E_{MC_j}$ is the amount of energy that MC_j can receive until it is fully charged.

Between two adjacent Special Chargers the one with the higher energy supplies takes the other's boundary Mobile Charger in its group. Thus, the Special Charger with lower energy supplies is responsible for a smaller area. In the case that their energy supplies are the same they do not exchange any Mobile Chargers. More precisely, the coordination algorithm is the following:

$\underline{(SC_k, SC_{k-1})}$

if $(e_k^p > e_{k-1}^n)$ **then**
$\mathcal{C}_k = \mathcal{C}_k \bigcup \{MC_{p_k}\}$
$\mathcal{C}_{k-1} = \mathcal{C}_{k-1} \setminus \{MC_{p_k}\}$
else if $(e_k^p < e_{k-1}^n)$ **then**
$\mathcal{C}_{k-1} = \mathcal{C}_{k-1} \bigcup \{MC_{n_{k-1}}\}$
$\mathcal{C}_k = \mathcal{C}_k \setminus \{MC_{n_{k-1}}\}$
else
No Mobile Chargers exchange
end if

$\underline{(SC_k, SC_{k+1})}$

if $(e_k^n > e_{k+1}^p)$ **then**
$\mathcal{C}_k = \mathcal{C}_k \bigcup \{MC_{n_k}\}$
$\mathcal{C}_{k+1} = \mathcal{C}_{k+1} \setminus \{MC_{n_k}\}$
else if $(e_k^n < e_{k+1}^p)$ **then**
$\mathcal{C}_{k+1} = \mathcal{C}_{k+1} \bigcup \{MC_{p_{k+1}}\}$
$\mathcal{C}_k = \mathcal{C}_k \setminus \{MC_{p_{k+1}}\}$
else
No Mobile Chargers exchange
end if

Trajectory: Special Charger k should determine which Mobile Charger will be the next that will be charged prioritizing a Mobile Charger based on minimum energy and minimum distance. Considering this, SC_k chooses to charge MC_m where

$$m = \arg\min_{j \in \mathcal{C}_k} \left\{ \left(1 + \frac{E_{MC_j}}{E_{MC}^{\max}}\right) \cdot \left(1 + \frac{d_{k_j}}{2R}\right) \right\}.$$

Charging phase: A Special Charger charges a Mobile Charger j according to its energy consumption rate r_{MC_j}. More specifically, a Mobile Charger with higher consumption rate (compared to the rest Mobile Chargers that belong to the Special Charger's group) should be charged with a higher amount of energy. Motivated by that, if by MC_m we denote the Mobile Charger that Special Charger k chose to charge, then the amount of energy that the Special Charger will give to it is $e = c_m \cdot \left(\min\{E_{MC_m}^{lack}, E_{SC_k}\}\right)$ where $c_m = \frac{r_{MC_m}}{\sum_{j \in \mathcal{C}_k} r_{MC_j}}$.

4.3 2-Level Knowledge Centralized Coordination 2KCC

The 2KCC algorithm performs centralized coordination and assumes 2-level network knowledge, i.e. it can use information both about Mobile Chargers' and about the sensors' energy status. It assigns to each Special Charger an amount of Mobile Chargers according to their residual energy. More precisely:
Coordination: $|C_k| = \mathcal{E}_k \cdot M$ where $\mathcal{E}_k = \frac{E_{SC_k}}{\sum_{i=1}^{S} E_{SC_i}}$, $(1 \leq k \leq S)$.
Trajectory: Since each Special Charger assumes 2-level network knowledge, it takes into account information from both Mobile Chargers and sensor nodes in order to find good trajectories. Thus, SC_k prioritizes MC_m where

$$m = \arg\min_{j \in \mathcal{C}_k} \left\{ \alpha \cdot \frac{E_{MC_j}}{E_{MC}^{\max}} + (1 - \alpha) \cdot \frac{\sum_{i \in \mathcal{D}_j} E_i}{|\mathcal{D}_j| \cdot E_{sensors}^{\max}} \right\}$$

with $\alpha \in (0, 1)$ a constant allowing to select the weight of each term in the sum. We use network lifetime (one of the most indicative performance metrics) to decide which is the appropriate value of parameter α in 2KCC protocol that achieves the best performance. As shown in Fig. 2 the most suitable value is $\alpha = 1$, which is explained by the fact that when a Special Charger charges a Mobile Charger, it should take into account its energy status only.

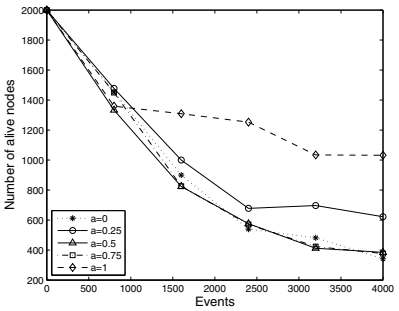


Fig. 2. Alive nodes over time (Varying α)

Charging phase: Each Special Charger computes the percentage of energy to transfer, according to the lack of energy in the Slice of the selected Mobile Charger compared to the total energy lack in all Slices that this Special Charger is responsible for. More precisely, Special Charger k transfers to MC_m an amount of energy $e = c_m \cdot \left(\min\{E^{lack}_{MC_m}, E_{SC_k}\}\right)$ where

$$c_m = \frac{\sum_{i \in \mathcal{D}_m} E_i^{lack}}{\sum_{j \in \mathcal{C}_k} \sum_{i \in \mathcal{D}_j} E_i^{lack}} \quad \in (0, 1)$$

where $E_i^{lack} = E_{sensor}^{\max} - E_i$ is the amount of energy that sensor i can receive until it is fully charged.

References

1. Angelopoulos, C.M., Nikoletseas, S., Raptis, T.P.: Wireless energy transfer in sensor networks with adaptive, limited knowledge protocols. Computer Networks **70**, 113–141 (2014)
2. Angelopoulos, C.M., Nikoletseas, S., Raptis, T.P., Raptopoulos, C., Vasilakis, F.: Improving sensor network performance with wireless energy transfer. International Journal of Ad Hoc and Ubiquitous Computing (in press, 2014)
3. Dai, H., Liu, Y., Chen, G., Wu, X., He, T.: Safe charging for wireless power transfer. In: Proceedings of the 33rd Annual IEEE International Conference on Computer Communications, INFOCOM (2014)
4. Dai, H., Wu, X., Chen, G., Xu, L., Lin, S.: Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks. Computer Communications **46**, 54–65 (2014)
5. Madhja, A., Nikoletseas, S., Raptis, T. P.: Efficient, distributed coordination of multiple mobile chargers in sensor networks. In: Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM (2013)
6. Wang, C., Li, J., Ye, F., Yang, Y.: Netwrap: An NDN based real-time wireless recharging framework for wireless sensor networks. IEEE Transactions on Mobile Computing **13**(6), 1283–1297 (2014)